



Digital House

★ Favorites	<input type="checkbox"/>
📌 Tags	
🕒 Last edited time	@30 de junio de 2025 15:40
🕒 Created time	@30 de junio de 2025 13:04
⚙️ Status	Sin empezar

Testing

¿Cómo escribir un caso de prueba?

Un **caso de prueba** debe contener:

1. **ID del Caso de Prueba:** Ej. TC-REG-01
2. **Título:** Nombre breve que describa el objetivo.
3. **Funcionalidad relacionada:** Qué parte del sistema prueba.
4. **Precondiciones:** Lo que se necesita antes de ejecutar.
5. **Pasos para ejecutar:** Acciones detalladas a realizar.
6. **Datos de entrada:** Si aplica.
7. **Resultado esperado:** Qué debería ocurrir.
8. **Resultado obtenido:** (se llena durante la ejecución)
9. **Estado:** (Pass / Fail)

Ejemplo:

ID	TC-REG-01
Título	Registro exitoso de usuario
Funcionalidad	Registro
Precondición	No estar registrado

Pasos	1. Ingresar datos válidos 2. Presionar "Registrarse"
Datos	Email: válido Contraseña: válida
Resultado Esperado	Redirige a pantalla de Login con mensaje de éxito
Resultado Obtenido	(se completa al ejecutar)
Estado	(Pass/Fail)

¿Cómo reportar un defecto?

Para reportar un defecto:

1. **ID del Bug:** BUG-001
2. **Título:** Breve descripción del error.
3. **Descripción:** Explicación detallada.
4. **Pasos para reproducir:** Cómo se genera el error.
5. **Resultado esperado:** Qué debería suceder.
6. **Resultado actual:** Qué está pasando realmente.
7. **Severidad:** (Crítica, Alta, Media, Baja)
8. **Adjuntos:** Capturas de pantalla, videos o logs.
9. **Estado del bug:** Nuevo / Asignado / Cerrado / Reabierto

Ejemplo:

ID	BUG-002
Título	Error al registrarse con email válido
Descripción	El sistema devuelve error 500 al usar email válido.
Pasos	1. Ir a Registro 2. Ingresar email válido 3. Enviar
Resultado Esperado	Redirigir a login
Resultado Actual	Error 500
Severidad	Alta
Estado	Nuevo

Criterio para incluir en Suite de Smoke Test:

- Casos básicos que validan que el sistema es **estable**.

- Verifica que las funcionalidades críticas están operativas.
- Ejemplos:
 - Acceso al sitio en desktop, tablet y móvil.
 - Carga de la página de inicio.
 - Login con credenciales válidas.
 - Registro funcional básico.

Criterio para incluir en Suite de Regression Test:

- Casos que validan que **no se rompió nada** tras una modificación.
- Abarcan escenarios **positivos y negativos**.
- Se actualiza con cada nueva funcionalidad.
- Ejemplos:
 - Validaciones de email incorrecto.
 - Redirecciones post-registro/login.
 - Comprobación de mensajes de error.
 - Prueba con campos vacíos.

Testing Manual

Módulo: Página de Inicio

- TC-INI-01: Visualización correcta de imagen y texto desde DB
- TC-INI-02: Redirección a Login desde botón "Iniciar sesión"

Módulo: Registro

- TC-REG-01: Registro con datos válidos
- TC-REG-02: Registro con email inválido
- TC-REG-03: Registro con contraseña corta
- TC-REG-04: Registro con campos vacíos
- TC-REG-05: Redirección a login al registrarse

Módulo: Login

- TC-LOG-01: Login con credenciales válidas
- TC-LOG-02: Login con email inválido
- TC-LOG-03: Login con contraseña incorrecta
- TC-LOG-04: Redirección al /home tras login
- TC-LOG-05: Link a registro funcional

Epic: Dashboard

TC-DASH-01 – Mostrar saldo disponible

- Validar que se muestren los dos centavos.
- Verificar formato ARS (\$X.XX)

TC-DASH-02 – Botón "Ingresar dinero"

- Verifica que el botón sea visible y redireccione correctamente (si aplica)

TC-DASH-03 – Menú lateral funcional

- Verifica visibilidad del menú en todas las pantallas
- Clic en nombre redirige al Dashboard

TC-DASH-04 – Últimos movimientos

- Se visualizan 10 resultados ordenados por fecha
 - Clic en "Ver toda la actividad" redirige (o queda inactivo, si no está implementado)
 - Campo búsqueda permite entrada y simula redirección futura
-

Epic: Perfil

TC-PER-01 – Visualización de datos personales

- Comprobar que se muestren correctamente nombre, alias, CVU

TC-PER-02 – Edición de datos y alias

- Verificar formato del alias (X.X.X)
- Guardar cambios y validar persistencia

TC-PER-03 – Copiado al portapapeles

- Validar copia de CVU y alias

TC-PER-04 – Gestión de medios de pago

- Clic en botón redirige correctamente a "Gestión de medios de pago"
-

Epic: Gestión de medios de pago

TC-MP-01 – Alta de tarjeta

- Clic en botón redirige a alta
- Validar que no se puedan registrar más de 10 tarjetas
- Mostrar mensaje al llegar al límite

TC-MP-02 – Detección automática del tipo de tarjeta

- Ingresar primeros 4 dígitos y validar tipo (Visa, Mastercard, Amex)

TC-MP-03 – Listado de tarjetas

- Mostrar últimos 4 dígitos
- Mostrar todas las tarjetas vinculadas

TC-MP-04 – Eliminación de tarjeta

- Al eliminar la última tarjeta, mostrar mensaje "No tienes tarjetas asociadas"
-

Epic: Cerrar sesión

TC-LOGOUT-01 – Token en LocalStorage

- Validar que **no se elimine** el token al recargar la página

TC-LOGOUT-02 – Cierre de sesión

- Al hacer clic: se elimina token, se redirige a página promocional

Clasificación en Smoke y Regression:

Smoke Test Suite

- TC-INI-01
- TC-INI-02
- TC-REG-01
- TC-LOG-01

- TC-DASH-01 – Ver saldo
- TC-DASH-03 – Menú lateral visible y funcional
- TC-PER-01 – Ver datos del perfil
- TC-MP-01 – Alta de tarjeta
- TC-LOGOUT-02 – Cerrar sesión correctamente
- ST-01 : Verificar carga de medios de pago disponibles
- ST-02:Verificar ingreso de monto y pantalla resumen
- ST-03:Visualización de CBU y Alias
- ST-04:Carga de la actividad reciente del usuario
- *TC-SERVICE-01*:Visualización de servicios
- TC-SERVICE-02:Búsqueda por título de servicio
- TC-SERVICE-03:Ingreso de número de cuenta inválido
- TC-SERVICE-04:Ingreso de número de cuenta válido
- TC-SERVICE-05:Selección de medio de pago existente
- TC-SERVICE-06:Agregar nuevo medio de pago
- TC-SERVICE-07:Pago con dinero en cuenta (fondos suficientes)
- TC-SERVICE-08:Pago con dinero en cuenta (fondos insuficientes)

Regression Test Suite

Inicio

- **TC-INI-01:** Visualización de texto e imagen desde la base de datos.
- **TC-INI-02:** Redirección desde botón "Iniciar sesión".
- **TC-INI-03:** Responsive en desktop, tablet y mobile.

Registro

- **TC-REG-01:** Registro exitoso con datos válidos.
- **TC-REG-02:** Error con email inválido.
- **TC-REG-03:** Validación de contraseña débil.
- **TC-REG-04:** Mensaje de error con campos vacíos.

- **TC-REG-05:** Redirección a login tras registro.

Login

- **TC-LOG-01:** Login exitoso con email y contraseña válidos.
- **TC-LOG-02:** Error con email incorrecto.
- **TC-LOG-03:** Error con contraseña inválida.
- **TC-LOG-04:** Redirección a /Home tras login.
- **TC-LOG-05:** Link de registro funcional desde login.

Dashboard

- **TC-DASH-01:** Visualización de saldo con dos centavos (ARS).
- **TC-DASH-02:** Acceso al botón "Ingresar dinero".
- **TC-DASH-03:** Menú lateral visible en todas las páginas.
- **TC-DASH-04:** Redirección al Dashboard desde nombre de usuario.
- **TC-DASH-05:** Mostrar últimos 10 movimientos, ordenados.
- **TC-DASH-06:** Simulación de búsqueda (aunque aún no funcione).
- **TC-DASH-07:** Botón "Ver toda la actividad" presente.

Perfil

- **TC-PER-01:** Visualización de nombre, CVU, alias, contraseña oculta.
- **TC-PER-02:** Validación del formato del alias al editar (X.X.X).
- **TC-PER-03:** Persistencia de datos modificados al guardar.
- **TC-PER-04:** Copiar alias y CVU al portapapeles.
- **TC-PER-05:** Redirección al gestionar medios de pago.

Gestión de medios de pago

- **TC-MP-01:** Alta de tarjeta redirige correctamente.
- **TC-MP-02:** Máximo de 10 tarjetas. Validación de límite.
- **TC-MP-03:** Detección automática de tipo de tarjeta (Visa, etc.)
- **TC-MP-04:** Mostrar lista de tarjetas (solo últimos 4 dígitos).
- **TC-MP-05:** Mensaje "No tienes tarjetas asociadas" si se elimina la última.

Cerrar sesión

- **TC-LOGOUT-01:** Token **no** se elimina al recargar el navegador.
- **TC-LOGOUT-02:** Al cerrar sesión, token se elimina y redirige a la página promocional.

Actividad

- **TC-ACTIVITY-01:**Filtro por período en historial de transacciones
- **TC-ACTIVITY-02:**Filtro por tipo de operación (ingresos/egresos)
- **TC-ACTIVITY-03:**Paginación del historial
- **TC-ACTIVITY-04:**Búsqueda por palabra clave en transacción
- **TC-ACTIVITY-05:**Ver detalle de transacción individual

Pago de Servicios

- **TC-SERVICE-01:**Validación de flujo completo de pago
- **TC-SERVICE-02:** Listado de medios de pago
- **TC-SERVICE-03:** Validación con endpoint `account` para saldo disponible
- **TC-SERVICE-04:**Persistencia de selección de medio de pago

Testing Exploratorio

Se realizó un **test exploratorio estructurado** para detectar comportamientos no previstos y verificar la usabilidad general. Se utilizó un enfoque basado en **Sesiones, Tours, y Workflows**.

Organización del Test Exploratorio (Sprint III)

1. Sesiones Exploratorias

- Duración: 30 minutos por sesión
- Participantes: Dev
- Herramienta usada: Notion + Captura de pantalla (Snagit)

Sesión	Objetivo	Observaciones
--------	----------	---------------

S1	Validar ingreso con tarjeta	Se validó correcto flujo de ingreso hasta pantalla resumen
S2	Revisión de filtros de actividad	Buen desempeño, sin errores al aplicar o quitar filtros
S3	Detalles de transacciones	Se detectó leve retardo en carga de detalle, reportado a backend
S4	Copiado de CBU y Alias	Copiado funcional, falta retroalimentación visual ("copiado")

2. Tours Utilizados

- **Freestyle Tour:** Se navegó libremente por toda la sección de "Mi Actividad" sin filtros para observar comportamientos anómalos.
- **Scenario Tour:** Simulación de un flujo completo desde ingreso de dinero hasta ver su reflejo en la actividad.
- **Data Tour:** Revisión con transacciones masivas (>100) para verificar paginación, rendimiento y consistencia.

3. Workflows Verificados

- Ingreso de dinero → Confirmación → Actividad registrada
- Visualización de CBU/alias → Copiado → Pega en app externa
- Filtrado + búsqueda + reinicio filtros → Comportamiento esperado

Hallazgos Relevantes

- El botón de "copiar" no tiene feedback visual. Se recomendó agregar un mensaje como "Copiado al portapapeles".
- En dispositivos con baja conectividad, el detalle de transacción puede demorar >2 segundos en aparecer.
- Todo el resto del flujo es funcional, consistente y sin errores críticos.

Organización del Test Exploratorio (Sprint IV)

1. Sesiones Exploratorias

Sesión	Objetivo	Duración	Observaciones
--------	----------	----------	---------------

S1	Navegación por listado y búsqueda de servicios	30 min	Búsqueda rápida, se validó comportamiento sin resultados
S2	Ingreso de número de cuenta (válido/inválido)	30 min	Validaciones funcionales. Se simulon errores manuales
S3	Selección y adición de medios de pago	30 min	Flujo intuitivo. Se validó persistencia al retroceder
S4	Validación de pagos (con y sin fondos)	30 min	Integración con endpoint <code>account</code> funcional, feedback claro

2. Tours Utilizados

- **Data Tour:** Pruebas con múltiples tipos de cuentas y saldos distintos
- **Error Tour:** Ingreso intencional de datos inválidos para validar mensajes de error y validaciones
- **Scenario Tour:** Se simulon flujos completos desde búsqueda de servicio hasta confirmación de pago

3. Escenarios Probados

- Servicio inexistente → Ingreso de número inválido → Error correcto
- Servicio válido → Número válido sin deuda → Mensaje sin facturas
- Servicio válido → Número válido con deuda → Medio de pago → Confirmación
- Selección de pago con dinero en cuenta → Sin fondos → Error mostrado
- Selección de pago con tarjeta nueva → Agregado exitoso → Pago correcto

4. Workflows Verificados

- Buscar servicio → Ingresar número de cuenta → Ver error → Reintentar con éxito
- Selección de servicio → Medio de pago nuevo → Pago correcto
- Servicio → Medio de pago → Pago fallido por fondos → Cambio de medio de pago → Éxito

Hallazgos Destacados

- UX: Falta de retroalimentación inmediata tras ingresar número válido; se recomienda spinner o "validando..."
- Validaciones por saldo (endpoint `account`) no integrado. Falta diseño de UI.
- Flujo completo de pago funciona correctamente en la mayoría de los escenarios.

Infraestructura

Componentes Principales

Componente	Descripción
Next.js App	Aplicación frontend (SSR/SSG) + API endpoints internos
Strapi CMS	Backend CMS que proporciona contenido mediante REST o GraphQL
Redis	Almacenamiento temporal de tokens u otra info de sesión/cache
Base de Datos	Usada por Strapi (PostgreSQL, MySQL, o MongoDB, dependiendo de la config)
Servicio Externo	API de terceros al que Next.js se conecta mediante endpoints internos
Load Balancer (opcional)	Balanceo de carga para escalabilidad de frontend y backend
CDN (ej: Vercel o Cloudflare)	Distribución de contenido estático

Despliegue Sugerido

Componente	Entorno sugerido
Next.js	Vercel, Netlify o contenedor Docker
Strapi	VPS o Docker en ECS/EC2/K8s
Redis	Redis Cloud, Upstash o instancia EC2
DB	RDS (AWS), PlanetScale, Supabase
CDN	Vercel CDN, Cloudflare, Fastly
External API	Expuesta como REST/GraphQL, con autenticación adecuada

Consideraciones de Seguridad y Escalabilidad

- Usa **HTTPS** en toda la red
- Protege el Redis con ACL y red privada
- Los secretos y tokens deben almacenarse en **env vars**
- Implementa **rate limiting** en tus endpoints

Boceto de red

